



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/690,818	10/23/2003	Patrick J. Link	723-1443	5437
27562 7590 01/22/2009 NIXON & VANDERHYE, P.C. 901 NORTH GLEBE ROAD, 11TH FLOOR ARLINGTON, VA 22203				
EXAMINER				
DUFFY, DAVID W				
ART UNIT		PAPER NUMBER		
3714				
MAIL DATE		DELIVERY MODE		
01/22/2009		PAPER		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents  
United States Patent and Trademark Office  
P.O. Box 1450  
Alexandria, VA 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

**BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES**

Application Number: 10/690,818  
Filing Date: October 23, 2003  
Appellant(s): LINK, PATRICK J.

---

Robert Faris  
For Appellant

**EXAMINER'S ANSWER**

This is in response to the appeal brief filed 10/16/2008 appealing from the Office action mailed 01/31/2008.

**(1) Real Party in Interest**

A statement identifying by name the real party in interest is contained in the brief.

**(2) Related Appeals and Interferences**

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

**(3) Status of Claims**

The statement of the status of claims contained in the brief is correct.

**(4) Status of Amendments After Final**

No amendment after final has been filed.

**(5) Summary of Claimed Subject Matter**

The summary of claimed subject matter contained in the brief is correct.

**(6) Grounds of Rejection to be Reviewed on Appeal**

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

**(7) Claims Appendix**

The copy of the appealed claims contained in the Appendix to the brief is correct.

**(8) Evidence Relied Upon**

Snes9x: The Portable Super Nintendo Entertainment System Emulator v1.19 readme.txt

Gameboy98

Z80-68K-v150 Z80 Engine written in 68020 assembler for inclusion in C/C++ projects

US 6658056 B1	Duruoz; Cem I. et al.	12-2003
US 20030207712 A1	Sterchi, Henry C et al.	11-2003
US 6192446 B1	Mullarkey; Patrick J. et al.	02-2001
US 6084600 A	Munshi; Aaftab A.	07-2000
US 6058288 A	Reed; Danny Paul et al.	05-2000
US 6007428 A	Nishiumi; Satoshi et al.	12-1999
US 5949985 A	Dahl; Stephen A. et al.	09-1999
US 5903760 A	Farber; Yaron et al.	05-1999
US 5790825 A	Traut; Eric P.	08-1998
US 5781758 A	Morley; John E.	07-1998
US 5153577 A	Mackey; Kevin J. et al.	10-1992
US 4771279 A	Hannah; Marc R.	09-1988

**(9) Grounds of Rejection**

The following ground(s) of rejection are applicable to the appealed claims:

Claims 6, 14, 18, 23, 27, 29, 32 and 35-58 and are rejected under 35 U.S.C. 103(a) as being unpatentable over Snes9x: The Portable Super Nintendo Entertainment System Emulator v1.19 readme.txt dated June 5, 1999 in view of Dahl et al (US 5,949,985) and further in view of Gameboy98.

Regarding claims 6 and 14, Snes9x: The Portable Super Nintendo Entertainment System Emulator v1.19 (herein referred to as "Snes9x") discloses an emulator program that emulates the SUPER NINTENDO ENTERTAINMENT SYSTEM (SNES). This emulator program is executed by a target platform such as a PC that is different than the video game platform (Snes9x page 5). It is well known in the art that a PC is capable of displaying graphical information on a target computing display device that has read/write memory and is capable of receiving user inputs. The Snes9x emulator processes and parses SNES ROM images (Snes9x page 6). The emulated platform is a handheld platform in that a player holds the game controllers of a SNES game system in their hands, thus constituting a handheld video game system. Snes9x also discloses modeling at least some display timing activities of the handheld video game device (page 8). Snes9x is an emulator program that emulates videogame software initially intended only for use with the SNES and so the videogame software is still capable for use with the original system. The Snes9x emulator produces real time interactive game presentations on the target platform (Snes9x page 4). The Snes9x emulator allows the

target platform the option of running the emulated games in a windowed mode or a full screen mode (Snes9x page 7).

Snes9x is silent regarding the ROM pages and said method further includes said emulator program allocating ROM pages in said target computing device read/write memory and duplicating at least a portion for said allocated ROM pages. Dahl et al. discloses a method and data processing system for emulating a program that utilizes a typical paging architecture that swaps the pages between a main store and a DASD. To make the process more efficient and minimize overhead during emulation, the data is accessed identically, which minimizes paging. Since the emulated DASD and emulated main store are accessed identically, it is reasonable to assume that they contain the same files. The difference is that upon execution, the files in the main store, which are often scattered in discontinuous physical locations, are copied into an emulated DASD that is stored continuously (abstract, col. 2, lines 45-55, col. 5, lines 15-65). Furthermore, Dahl et al. discloses caching of instruction portions of code (col. 6, lines 1-15). Caching is well known in the art as duplicating a portion of data stored in the system, which is frequently used in order to minimize the resources by eliminating the need to constantly access the code every time it is needed. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to utilize the paging architecture and duplication of ROM pages in order to minimize address translation and paging overhead during emulation (col. 5, lines 49-52).

It should also be noted that the combination of Snes9x and Dahl would create a game system that has a pointer table system that performs the functions as claimed by

the applicant. Detailed discussion of the functions of the pointer table system can be found above.

The prior art is further silent regarding emulation of a handheld video game device wherein the housing, and display are in one piece and runs video game software image out of a page-based read only memory (ROM). Gameboy98 teaches the emulation of the Nintendo's GAMEBOY system which is a handheld system as described above using ROM cartridges (pages 1-4). Therefore it would have been obvious to one of ordinary skill in the art at the time of applicant's invention to emulate the games of Nintendo's GAMEBOY system since GAMEBOY is a very popular system with many fans and favored games.

At the time the invention was made, it would have been an obvious matter of design choice to a person of ordinary skill in the art to duplicate commonly used instructions (ROM pages) across a set of emulated instructions because applicant has not disclosed that duplicating of ROM pages across said ROM pages provides an advantage, is used for a particular purpose, or solves a stated problem. One of ordinary skill in the art, furthermore, would have expected Dahl's duplication of instructions on a cache memory, and applicant's invention, to perform equally well with either the duplicated ROM pages of the claim or the cached instructions of Dahl because both duplications perform the same function of reducing page swapping. Therefore, it would have been prima facie obvious to modify the prior art to obtain the invention as specified in claim 1 because such a modification would have been considered a mere design consideration which fails to patentably distinguish over the prior art.

Regarding claim 18, the prior art is further silent regarding modeling the liquid crystal display of the handheld game system using a virtual liquid crystal display controller to maintain real time synchronization. Gameboy98 teaches the use of such a LCD display controller (page1 dated Dec 1999, page 2 11<sup>th</sup> Nov 1998) to emulate the effects of the GAMEBOY LCD screen on the target computing device that may not have the same LCD display as that of the GAMEBOY system. Therefore it would have been obvious to one of ordinary skill in the art at the time of applicant's invention to include in the emulation system an LCD controller as that taught by Gameboy98 since it is known that the GAMEBOY system had a unique LCD display that requires a controller to adapt its functions to a target computing display that may not have the same functionality as that of the GAMEBOY's display. Such an adaptation and addition would preserve the game play of the GAMEBOY games on the target emulation system.

Regarding claim 23, Snes9x does not disclose specifically the use of a page table to remap memory access instructions into different memory locations. However, Dahl et al. discloses a system wherein an emulator program may be used to emulate a target platform on a different host platform. This emulation system that Dahl et al. discloses utilizes translation or page tables that associates page selector bits with a 16-bit page number, which is concatenated to the page offset to access appropriate physical memory locations (Dahl 5:11 – 43). One of ordinary skill in the art would be motivated to modify Snes9x in view of Dahl et al. to provide an emulation system that utilizes page tables for the remapping of memory access instructions into different memory locations. One would be motivated to do so because Dahl states that by

referencing storage objects through a translation table, the memory pages associated with a storage object can be scattered in discontinuous memory locations within the main store, while appearing to be contiguous.

Regarding claim 27, Snes9x discloses a frame skip count that enable the selectively skip frames (Snes9x page 8).

Regarding claim 29, Snes9x discloses a video game platform emulator as discussed above. The video game emulator emulates a video game platform on a different target platform. The purpose of the emulator is to emulate the video game platform on the target platform in every way. This would include the video game platforms memories and registers, including the size of the registers, such as byte, word and long formats. Thus it would be obvious to create an emulator that emulates these aspects of the video game platform on the target platform. One of ordinary skill in the art would be motivated to modify Snes9x to emulate the memory structures of the video game platform, wherein the formats include byte, word and long register formats. One would be motivated to do so because these are common register formats and purpose of the emulator is to emulate all aspects of the video game platform on the target platform, thus emulating the register formats.

Regarding claim 32, Snes9x discloses that a wide variety of joysticks or user input devices can be used by the Snes9x emulator (Snes9x page 10).

Regarding claims 35 and 36, Snes9x is for use in PCs, however it is well known in the art that laptop computers possess the same functionality and performance characteristics, which would make them capable of performing the same tasks as a

standard desktop PC. The fact that they are laptops makes them handheld and portable. Furthermore, the examiner does not believe it to be a patentably distinct feature to implement the emulation program into a PDA since it is well known in the art that computers are constantly getting smaller and that devices such as PDAs and cellular phones are constantly being given more and more functionality of computers. Therefore it would be a matter of routine to one of ordinary skill in the art at the time of invention to implement the emulation program of Snes9x into a PDA or handheld portable computer in order to allow for users to play the games while being mobile.

Regarding claim 37, the limitations of claim 37 have been addressed above, Regarding the preamble which requires a handheld portable battery-operated computer device, please refer to the discussion regarding claims 35 and 36 above.

Regarding the image software, ROM pages, generation of real time audio-visual presentation in response to video game software image, displaying audio-visual image as a subset of display area, and duplicating ROM pages, please see the discussion above regarding claims 6 and 14.

Regarding claims 38, Snes9x emulates the Super Nintendo game system that emulates the ROM-based pluggable game cartridges of the system.

Regarding claims 39, 46 and 53, the prior art is silent regarding explicitly reserving at least twice the space occupied by the ROM pages and duplicating half of each page allocated in random access memory. As discussed above in claim 6, Dahl teaches the use of emulating program data and duplication of instructions, however Dahl does not teach the explicit "half of each page allocated in random access

memory." At the time the invention was made, it would have been an obvious matter of design choice to a person of ordinary skill in the art to duplicate the ROM pages as claimed because applicant has not disclosed that duplicating the pages at least twice the space occupied by the ROM pages and duplicating half of each page allocated in random access memory provides an advantage, is used for a particular purpose, or solves a stated problem. One of ordinary skill in the art, furthermore, would have expected the instruction duplication of Dahl and applicant's claimed invention to perform equally well with either the paging architecture as taught by Dahl or the claimed allocating at least twice the space occupied by ROM pages in the handheld video game device and duplicating half of each page allocated in random access memory. Therefore it would have been a prima facie obvious to modify the prior art to obtain the invention as claimed by claim 39 because such a modification would have been considered a mere design consideration which fails to patentably distinguish over the prior art.

Regarding claims 40, 47 and 54, Snes9x teaches the use of emulation of cartridge ROM. However Snes9x does not explicitly require that the ROM banks are emulated in a different RAM page. Dahl teaches the use of RAM pages for each set of instructions (ROM bank) (5:1-6:67). Therefore it would have been obvious to one of ordinary skill in the art at the time of invention to segregate the ROM banks into separate RAM pages in order to break them down to more manageable chunks of data. Such a modification would make the system operate more smoothly, since the

processing of bits or commands/instructions one at a time requires less processing power than that of processing an entire program.

Regarding claims 41, 48 and 55, please see rejection of claim 6 and 40 above. Regarding the limitations of a page count register, it is notoriously old and well known in the art that all paging architectures require the use of a page count register to function since the computer needs to know which page was previous loaded or has been loaded so that in the event that a previously loaded page is required, it may page swap to that page. Such limitations are required by said systems as with out such limitations of a selection pointer, the computer would not know where the required commands are stored in memory (Dahl 5:11-46). Therefore it would have been notoriously obvious to one of ordinary skill in the art to use a page count register as required by claim 41 to have an operating page architecture since without the count register, the system would not know which pages have been loaded and would not be able to swap between loaded pages and therefore defeat the purpose of a paging architecture since without the count register or some means of indicating a page has been loaded, the system would not work.

Regarding claims 42-45, 49-52 and 56-58, the prior art fails to explicitly teach a write protect function on the emulated ROMs. However it is notoriously old and well known in the art to write protect data that one does not wish to be altered such data would include that of the ROM images since they are originally intended to be Read-Only-Memory and therefore an emulation of such would require the emulated ROM pages to be write-protected as well. Write protecting data is a mater of routine to one of

ordinary skill in the art and therefore it would have been obvious to one of ordinary skill in the art at the time of invention to write protect the ROM data files in order to prevent the data from being altered since an alteration of the ROM files could possibly render the game useless.

Claim 17 is rejected under 35 U.S.C. 103(a) as being unpatentable over Snes9x: The Portable Super Nintendo Entertainment System Emulator v1.19 in view of Dahl et al. and GameBoy98 as applied to claim 6 above and further in view of Nishiumi et al (US 6,007,428).

Regarding claim 17, Snes9x is silent as to what type of display screen that the system can utilize, such as a liquid crystal display. However, Nishiumi et al discloses a video game system wherein the display unit is a liquid crystal display (Nishiumi et al 25:56 – 57).

It would be obvious to one of ordinary skill in the art to modify Snes9x in view of Nishiumi et al for the purpose providing a video emulator system that displays emulated video information on a liquid crystal display. One would be motivated to use a liquid crystal display because they utilize less space and provide accurate video representations.

Claim 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over Snes9x: The Portable Super Nintendo Entertainment System Emulator v1.19 readme.txt in view

of Dahl et al. and Gameboy98 as applied to claim 6 above and further in view of Munshi (US 6,084,600).

Regarding claim 19, Snes9x discloses a video game platform emulator. Snes9x does not disclose the use of hardware assisted BLIT memory transfers. However Munshi et al discloses a system for the speeding up of pixel data transfers (bitblits) by compressing and word aligning the data transferred. Munshi discloses that one way to reduce the amount of required bandwidth in a system when transferring pixel data is to use bitblit operations, wherein a rectangular region within the display memory is specified and data for pixels within the region is transferred (Munshi 1:57 – 62). While, Munshi discloses that this approach is sometimes associated with problems, it is stated that this is still effective for transferring data.

One of ordinary skill in the art would be motivated to modify Snes9x in view of Munshi for the purpose of using Bitlit operations to transfer pixel data. BLIT operations are known for reducing the bandwidth loads upon a system when updating pixel information in memory.

Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over Snes9x: The Portable Super Nintendo Entertainment System Emulator v1.19 readme.txt in view of Dahl et al. and Gameboy98 as applied to claim 6 above and further in view of Mackey et al (US 5,153,577).

Regarding claim 20, Snes9x does not disclose a recomputed translation table that translates the graphics of the video game platform. However, Mackey et al

discloses an emulation system that will emulate a PC on another host system or platform (Mackey et al 3:51 – 55). The system of Mackey et al. discloses the use of translation tables (Tables 1, 6 – 9) that determine how that graphical character formats will be displayed upon the host platform (Mackey 44:60 – 45:60).

One of ordinary skill in the art would be motivated to modify Snes9x in view of Mackey et al to provide an emulation system that utilizes translation tables to convert graphic formats that are compatible with the target platform. This would be beneficial because when running an emulator on a target platform the emulator must be able to translate the native graphics of the video game platform into a format the target platform can display.

Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over Snes9x: The Portable Super Nintendo Entertainment System Emulator v1.19 readme.txt in view of Dahl et al. and Gameboy98 as applied to claim 6 above and further in view of Sterchi (US 2003/0207712).

Regarding claim 21, the Snes9x system emulates the SNES video game platform. Thus, in order to run properly, the emulator must emulate or simulate the actual hardware and memory structures of the emulated system. Otherwise the original game ROMS would be rendered useless. Snes9x also discloses that in order for the emulator to run the target platform must have at least 16 MB of RAM (Snes9x page 5). While Snes9x does not specifically disclose that the emulator emulates registers and other hardware based memory structures in RAM, it is highly implied. However, Sterchi

et al discloses a system that uses an emulator, wherein the emulator can be run on an entirely different hardware platform than the video game platform. Sterchi also discloses that the emulator may emulate or simulate some or all of the hardware and/or software components of the target system (Sterchi page 2: par 22).

One of ordinary skill in the art would be motivated to modify Snes9x in view of Sterchi et al for the purpose of emulating registers and memory structures of the video game platform in RAM of the target platform. Sterchi discloses an emulator that would emulate all of the hardware or software components of the targets system, thus this would include components such as the registers and memory structures.

Claim 22 is rejected under 35 U.S.C. 103(a) as being unpatentable over Snes9x: The Portable Super Nintendo Entertainment System Emulator v1.19 readme.txt in view of Dahl et al. and Gameboy98 as applied to claim 6 above and further in view of Morley (US 5,781,758).

Regarding claim 22, Snes9x does not disclose the uses of a jump table to parse incoming binary instructions. Morley however does disclose an emulation system that utilizes jump tables for the purpose of reducing memory requirements (Morley 3:21 – 48). Morley discloses that a jump table is commonly referred to as a dispatch table (Morley 1:42 – 50). Morley discloses that instead of storing every instance of a routine, the dispatch table will store the routine in memory when it is called and also store the address of the routine in memory in place of the pointer, thus speeding up emulation.

One of ordinary skill in the art would be motivated to modify Snes9x in view of Morley to provide an emulation system that utilizes jump tables or dispatch tables to reduce the memory requirements on the system. By not storing every instance or a routine but only the address or a statically stored routine, unnecessary memory usage is eliminated.

Claim 24 is rejected under 35 U.S.C. 103(a) as being unpatentable over Snes9x: The Portable Super Nintendo Entertainment System Emulator v1.19 readme.txt in view of Dahl et al. and Gameboy98 as applied to claim 6 above and further in view of Z80-68K-v150 Z80 Engine written in 68020 assembler for inclusion in C/C++ projects, written by Gunter Woigk, dated Dec 25, 1999.

Regarding claim 24, Snes9x does not disclose a read only memory protection feature to prevent overwriting of ROM's during emulation. However, Woigk discloses an emulator package called the z80-68k which supports write protection for ROM's (Woigk page 2, 6).

One of ordinary skill in the art would be motivated to modify Snes9x in view of Woigk to provide an emulator that had write protection functions such that the ROM does not get overwritten during emulation. This would be of benefit due such that the ROMs are always protected from being overwritten.

Claims 25 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Snes9x: The Portable Super Nintendo Entertainment System Emulator v1.19

readme.txt in view of Dahl et al. and Gameboy98 as applied to claim 6 above and further in view of Hannah (US 4,771,279).

Snes9x teaches a CPU cycle timer that enables a user to set a percentage of CPU cycles to execute per can line. However, Snes9x fails to explicitly recite modeling including using a state machine defining at least a horizontal blank state and a vertical blank state. However in a related patent, Hannah teaches a method and apparatus for adapting high resolution images to low resolution monitors utilizing a multiple clock shift registers and HBLANK and VBLANK (horizontal and vertical blanks) states/signals (15:62-16:35). Hannah and Snex9x are analogous art because they both relate to adaptation of apparatus to different, conventionally non-adaptable, apparatus; such as high resolution signals to low resolution displays or game ROMs emulated to different game machines/systems. Therefore it would have been obvious to one of ordinary skill in the art at the time of applicant's invention to include the horizontal blank and vertical blank signals and clock system of Hannah with Snes9x so that the system can be emulated using a plurality of different display devices, making the game more adaptable to different display devices, including low resolution display devices.

Claim 28 is rejected under 35 U.S.C. 103(a) as being unpatentable over Snes9x: The Portable Super Nintendo Entertainment System Emulator v1.19 readme.txt in view of Dahl et al. and Gameboy98 as applied to claim 6 above and further in view of Mullarkey et al. (US 6,192,446).

Regarding claim 28, Snes9x does not disclose the use of a no-operation look-ahead feature. However, Mullarkey et al. discloses a no-operation look ahead feature that utilizes a look ahead circuit that will examine subsequent instructions and determine what banks in memory the instructions are intended for (Mullarkey 7:8 – 48). If the instructions are not meant for a specific bank of memory than that bank of memory can be powered down, thus optimizing the performance of the chip and system.

One of ordinary skill in the art would be motivated to modify Snes9x in view of Mullarkey et al. for the purpose of providing a no-operation look-ahead feature. This would provide for a more efficient system that optimizes chip performance. If instructions are not meant for specific memories, then those memories are not powered up, thereby saving chip processing time.

Claim 30 is rejected under 35 U.S.C. 103(a) as being unpatentable over Snes9x: The Portable Super Nintendo Entertainment System Emulator v1.19 readme.txt in view of Dahl et al. and Gameboy98 as applied to claim 6 above and further in view of Farber et al. (US 5,903,760).

Regarding claim 30, Snes9x discloses a video game platform emulator as discussed above. The video game emulator emulates a video game platform on a different target platform. The purpose of the emulator is to emulate the video game platform on the target platform in every way. This would include the way in which the emulated video game platform would handle CPU flags conditions. Snes9x does not specifically disclose the details of how the emulator handles CPU flags. However,

Farber et al et al discloses a system of running Reduced Instruction Set Architecture (RISC) code on a system that is meant for Complex Instruction Set Architecture (CISC) or vice versa. Farber discloses that RISC may not support the setting of status flags as CISC would (Farber 2:17 – 22). Thus, Farber discloses a method of executing code that utilizes the setting of status flags on a system that has no native support for status flags (Farber 2:64 – 67). Farber discloses the use of a translator that is akin to an emulator program in that both transform non-native code into a format that the target platform can utilize (Farber 3:42 – 54). Farber discloses that the translation capsule table emulates the setting of status flags as part of the translated compare function. The flags are emulated by allocating bit fields within a general-purpose register (Farber 4:60 – 5:3).

One of ordinary skill in the art would be motivated to modify Snes9x in view of Farber et al. for the purpose of modeling the emulation of the video game platform to allow for the updating of CPU flags, as this would enhance the capabilities of the target platform. The target platform would be able to preserve and emulate a video game platform that utilizes data such as flags even though the target platform may not natively do so.

Claim 31 is rejected under 35 U.S.C. 103(a) as being unpatentable over Snes9x: The Portable Super Nintendo Entertainment System Emulator v1.19 readme.txt in view of Dahl et al. and Gameboy98 as applied to claim 6 above and further in view of Traut (US 5,790,825).

Regarding claim 31, Snes9x discloses a video game platform emulator as discussed above. However, Snes9x does not explicitly disclose that the emulated video game platform program counter is mapped to the target platforms memory, such as in a general-purpose register. Traut discloses an emulation system, wherein the program counter of the guest or emulated platform is mapped to the cache of the host or target platform (Traut 4:17 – 29).

One of ordinary skill in the art would be motivated to modify Snes9x in view of Traut to provide a system that maps the program counter of the video game platform to the memory of the target system. This would provide for lower address translation overhead, thus eliminating burdensome demands on the target platform.

Claim 33 is rejected under 35 U.S.C. 103(a) as being unpatentable over Snes9x: The Portable Super Nintendo Entertainment System Emulator v1.19 readme.txt in view of Dahl et al. and Gameboy98 as applied to claim 6 above and further in view of Munshi (US 6,084,600) and Duruoz et al. (US 6,658,056 B1).

Regarding claim 33, Snes9x discloses a video game platform as discussed above. Snes9x discloses an emulator that can operate in full screen or windowed mode (Snes9x page 7). Snes9x does not disclose the use of screen memory buffers that are larger than the display are to improve paging usage and the transferring of a subset of said memory buffer using BITBLIT. Duruoz et al discloses a video graphics display method wherein a larger buffer memory is utilized for optimum use of the available memory and for optimal preservation of data to meet the highest performance

requirements (col. 11, lines 1-27). By using a larger memory buffer memory processing due to the constant rendering of the display area would be reduced. Thus display information would not have to be rewritten or transferred out of storage by means of paging. Munshi et al discloses a system for the speeding up of pixel data transfers (bitblits) by compressing and word aligning the data transferred. Munshi discloses that one way to reduce the amount of required bandwidth in a system when transferring pixel data is to use bitblit operations, wherein a rectangular region within the display memory is specified and data for pixels within the region is transferred (Munshi 1:57 – 62). While Munshi discloses that this approach is sometimes associated with problems, it is stated that this is still effective for transferring data.

One of ordinary skill in the art would be motivated to modify Snes9x in view of Munshi and Duruo et al. to provide a memory buffer that is larger than the display area and using BITBLT to transfer data from memory to the display area. This would enable the system to eliminate memory-processing overhead thus making m. BLIT operations are known for reducing the bandwidth loads upon a system when updating pixel information in memory.

Claim 34 is rejected under 35 U.S.C. 103(a) as being unpatentable over Snes9x: The Portable Super Nintendo Entertainment System Emulator v1.19 readme.txt in view of Dahl et al. and Gameboy98 as applied to claim 6 above, in view of Reed et al. (US 6,058,288).

Regarding claim 34, Snes9x discloses a video game platform as discussed above. Snes9x does not disclose the target platform comprising an airline seat-back controller. Reed et al. however discloses a passenger control unit that is in an airplane seat. The PCU can be in the armrest or the back of the seat (Reed et al 18:41 – 19:6). The airline system can be used for a plurality of uses such as gaming, computing or watching movies (Reed et al 6:26 – 36). Reed further states that the game system can be used for games like Super Nintendo Entertainment Service, which Snes9x emulates.

One of ordinary skill in the art would be motivated to modify Snes9x in view of Reed et al for the purpose of providing an emulator on a target platform such as an entertainment system on an airplane. This would provide passengers an enhanced flying experience despite being on long trips.

#### **(10) Response to Argument**

Appellant argues that Snes9x in view of Dahl and GameBoy98 does not teach or suggest duplicating at least a portion of said allocated emulated ROM pages across said ROM pages. Appellant's wording states that the emulated ROM pages are copied across the ROM pages of the video game software image which is contrary to the nature of a ROM being read only and serves no stated purpose. As best understood by the Examiner, appellant has intended to copy portions of the emulated ROM page across the emulated ROM page. By way of background, it is noted that the GAMEBOY system emulated by the current application utilizes a memory system with paged and fixed sections of memory space.

Snes9x copies the video game image ROM into the memory of the emulating system in order to function. Snes9x does not explicitly disclose copying at least a portion of the ROM pages across the ROM pages.

Dahl discloses a virtualization of a paging system for an emulated system whereby all of the data from a fixed and a paged section of storage for an emulated system is stored in active memory including the emulated paged memory to minimize address translation and reduce the paging overhead (5:46-65). The system of Dahl therefore teaches or suggests that by copying all portions of an emulated system's memory including paged storage into the active memory of a system the page swapping and address translation overhead of the emulating system will be reduced thus improving performance. The teachings of Snes9x and Dahl do not disclose emulating a GAMEBOY system.

GameBoy98 teaches that a GAMEBOY emulator was desirable and was known in the art at the time of appellant's invention. Accordingly, one of ordinary skill in the art would have been motivated to provide emulation of the GAMEBOY system in order to meet demand.

As the nature of the actual GAMEBOY system memory was that of a memory space composed of a fixed and a paged section, when emulating said system one of ordinary skill in the art would have been aware that the paging system would need to be emulated as well. One of ordinary skill in the art would then have recognized that only two options were present. The system could be emulated with a virtual paging system which would use less memory space by copying the video game ROM image as it was

originally configured, but such a configuration would require additional address translation for the emulating system to be able to access the virtualized paged sections and the emulated fixed portions of code because the virtualized pages would not be in the memory locations that were expected in relation to the fixed portions of code. Alternatively, the fixed sections of code could be copied for and included with each of the paged sections of memory to preserve the expected layout of the emulated system's memory space. Such a configuration would require more memory, but would reduce the address translation overhead by maintaining the expected arrangement of the memory space and reduce the amount of searching in memory to determine the location of paged and fixed sections. Given these two options and the suggestion by Dahl to reduce the address translation and memory swapping in order to improve performance, it would have been obvious to have copied the fixed portions of ROM data with the paged sections to preserve the expected layout of the emulated system memory space.

Appellant argues for claim 23 that the prior art does not teach or disclose a pointer table system to remap memory access instructions into function calls. Examiner respectfully disagrees. The disclosed system of Dahl provides a memory pointer table used to translate a memory function call into a translated memory function call.

Appellant argues for claim 29 that it would not have been obvious to model the instructions as a union of byte, word and long register formats. Examiner respectfully disagrees. A union is an old and well known computing data structure that specifically allows for the storage of a value that could take on several different but fixed types in

use since at least the 1960s. Given that the system is an emulator designed for use with many different types of emulating systems that may use different data types, it would have been obvious to one of ordinary skill that a union would allow any type of system to access the data without requiring costly duplication of code to accommodate all types of systems or additional overhead through increased translation of data types. Using a well known concept for its intended purpose is obvious and does not distinguish over the prior art.

Appellant argues for claims 39-41, 46-48 and 53-58 that the teachings of the prior art do not anticipate or render obvious allocating in RAM at least twice the space of the ROM pages and duplicating half of each page allocated in RAM. Again, appellant's wording of the claim is misleading. As best understood by the Examiner, the fixed portions of the video game image ROM are copied with the paged sections. If applicant were copying a portion of the paged data and duplicating said paged data the system would achieve no increase in performance and would not be enabled by appellant's specification. As such, Examiner uses the interpretation set forth above and contends that it would be obvious to allocate the necessary amount of RAM space to implement the emulation systems set forth above. One of ordinary skill in the art would have also recognized that in order to provide the emulation system utilizing the copied portions of code, more space would be needed in RAM, as explained above, to accommodate the extra data.

Appellant argues for claims 40, 47 and 54 that the prior art does not teach or suggest emulating ROM banks with a different RAM page. Appellant's wording is again

misleading. The common application of a paging system is to move data from an auxiliary store to RAM. The RAM itself does not have any pages; rather, the data is paged into and out of RAM from somewhere else. Thus, as best understood by Examiner from applicant's description, a "RAM page" would be the section of memory in the emulation system that holds the paged section and the fixed sections of data as one larger unit to maintain the original system's memory space arrangement. Accordingly, given the arrangement of the system taught by Snes9x, Dahl and GameBoy98 as discussed above it would have been obvious to maintain the arrangement of the original ROM pages when emulating the game in the system set forth above thereby creating the equivalent of appellant's "RAM pages".

Appellant argues for claims 42-45, 49-52 and 56-58 that the prior art does not teach or suggest a no-write function to protect emulated ROM space from being written to and alleges that Snes9x suggests to the contrary. Examiner respectfully disagrees. Examiner maintains that it would be extraordinarily obvious to write protect a game ROM from overwriting as otherwise the game would be altered and potentially ruined forcing the user to reload or reacquire data. Furthermore, the teachings of Snes9x mentioned by appellant do not suggest writing to game ROM space. As appellant is apparently unaware of the nature of the Game Genie, Action Replay and Gold finger systems; all the mentioned devices operate by modifying the game system's RAM values or by copying the ROM data and altering the copy, not by altering the game cartridge or the ROM data stored therein.

Appellant argues for claim 19 that Munshi criticizes "bitblit" and does not appear to disclose "hardware-assisted". Examiner respectfully disagrees. Munshi does disclose bitblit as a way to reduce bandwidth usage. Merely stating that such an option also has problems does not teach against its use or preclude the teachings of such a system as almost any system has a trade-off associated with it. Furthermore, every computer program is "hardware-assisted" as otherwise it would not exist or be operable.

Appellant argues for claim 28 that the prior art does not accomplish the same intended use as appellant's claim. Examiner respectfully disagrees. Mullarkey clearly teaches that look-ahead features may be used to optimize performance of a chip by powering down unneeded logic, powering up logic to reduce delay which Examiner contends is a teaching of avoiding wasted processing time waiting for the memory to charge.

**(11) Related Proceeding(s) Appendix**

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/David W Duffy/

Examiner, Art Unit 3714

Art Unit: 3714

Conferees:

/Dmitry Suhol/

Supervisory Patent Examiner, Art Unit 3714

/Greg Vidovich/

TQAS, TC 3700